

# Counting Subwords and Regular Languages

Charles J. Colbourn and Ryan E. Dougherty  
 Computing, Informatics, and Decision Systems Engineering  
 Arizona State University  
 P.O. Box 878809  
 Tempe, AZ 85287-8809  
 USA  
[ryan.dougherty@asu.edu](mailto:ryan.dougherty@asu.edu)  
[Charles.Colbourn@asu.edu](mailto:Charles.Colbourn@asu.edu)

Thomas Finn Lidbetter and Jeffrey Shallit  
 School of Computer Science  
 University of Waterloo  
 Waterloo, ON N2L3G1  
 Canada  
[finn.lidbetter@uwaterloo.ca](mailto:finn.lidbetter@uwaterloo.ca)  
[shallit@uwaterloo.ca](mailto:shallit@uwaterloo.ca)

June 22, 2018

## Abstract

Let  $x$  and  $y$  be words. We consider the languages whose words  $z$  are those for which the numbers of occurrences of  $x$  and  $y$ , as subwords of  $z$ , are the same (resp., the number of  $x$ 's is less than the number of  $y$ 's, resp., is less than or equal). We give a necessary and sufficient condition on  $x$  and  $y$  for these languages to be regular, and we show how to check this condition efficiently.

## 1 Introduction

A major theme in formal language theory is counting occurrences of letters in words. Let  $\Sigma$  be a finite alphabet. For a word  $z \in \Sigma^*$  and a letter  $a \in \Sigma$  we write  $|z|_a$  for the number of occurrences of  $a$  in  $z$ . A classic example of a language that is context-free but not regular

is  $\{z \in \{a, b\}^* : |z|_a = |z|_b\}$ ; see, for example, [4, Exercise 3.1 (e), p. 71]. The Parikh map (see, e.g., [7]) is another example of this theme.

We can generalize the counting of *letter* occurrences to the counting of *word* occurrences. Let  $w, y \in \Sigma^*$ . We say  $y$  is a *subword*<sup>1</sup> of  $w$  if there exist  $x, z \in \Sigma^*$  such that  $w = xyz$ . Define  $|w|_y$  to be the number of (possibly overlapping) occurrences of  $y$  in  $w$ . Thus, for example,  $|\text{banana}|_{\text{ana}} = 2$ .

In this paper we study the languages

$$\begin{aligned} L_{x < y} &= \{z \in \Sigma^* : |z|_x < |z|_y\}, \\ L_{x \leq y} &= \{z \in \Sigma^* : |z|_x \leq |z|_y\}, \\ L_{x = y} &= \{z \in \Sigma^* : |z|_x = |z|_y\} \end{aligned}$$

and their complements. The following is easy to see:

**Proposition 1.** *For all words  $x$  and  $y$ , the languages  $L_{x < y}$ ,  $L_{x \leq y}$ ,  $L_{x = y}$ , and their complements  $L_{x \geq y}$ ,  $L_{x > y}$ ,  $L_{x \neq y}$  are all deterministic context-free languages.*

*Proof.* We prove this only for  $L_{x = y}$ , with the other cases being analogous. We construct a deterministic pushdown automaton  $M$  that recognizes  $L_{x = y}$  as follows: its states record the last  $\max(|x|, |y|) - 1$  letters of the input seen so far. The stack of  $M$  is used as a counter to maintain the absolute value of the difference between the number of  $x$ 's seen so far and the number of  $y$ 's (a flag in the state records the sign of the difference). We have  $M$  accept its input if and only if this difference is 0. Since there is only one possible action for every triple of state, input symbol, and top-of-stack symbol,  $M$  is deterministic (any “invalid” configurations transition to a dead state  $d$ ).  $\square$

While  $L_{x = y}$  is always deterministic context-free, sometimes — perhaps surprisingly — it can also be regular. For example, when the underlying alphabet  $\Sigma$  is unary, then  $L_{x = y}$  is always regular. Less trivially, for  $\Sigma = \{0, 1\}$  it is an easy exercise to show that  $L_{01=10}$  is regular, and is recognized by the 5-state DFA in Figure 1; however,  $L_{01=10}$  is not regular when  $\Sigma = \{0, 1, 2\}$ . On the other hand,  $L_{0011=1100}$  is never regular, even when  $\Sigma = \{0, 1\}$ .

The goal of this paper is to give a necessary and sufficient condition for these languages to be regular, and show how to check it efficiently. The case  $L_{x < y}$  is covered in Section 5, and the case  $L_{x = y}$  is covered in Section 6. The remaining case  $L_{x \leq y}$  is virtually the same as  $L_{x < y}$ , and is left to the reader.

We assume a basic background in formal languages and automata; for all unexplained notions, see [4]. Four things are worth noting: if  $L$  is a language, then we write  $L^c$  for the complement  $\Sigma^* - L$ . Also, if  $x$  is a word, then  $x^R$  denotes the reverse of the word  $x$ . If  $x = a_1 a_2 \cdots a_n$  is a word, with each  $a_i \in \Sigma$ , then  $x[i..j] := a_i \cdots a_j$ . Finally, if  $x = tu$ , then by  $xu^{-1}$  we mean the word  $t$ , and by  $t^{-1}x$  we mean the word  $u$ .

---

<sup>1</sup>Sometimes called “factor”, especially in the European literature.

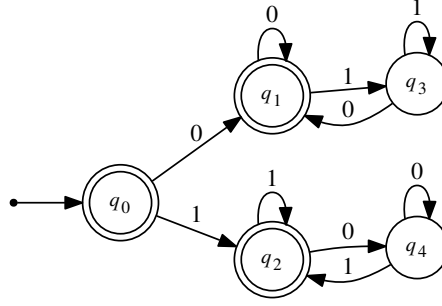


Figure 1: A DFA recognizing  $L_{01=10}$  over  $\Sigma = \{0, 1\}$

## 2 Bordered words and periodicity

Let  $y, z$  be words with  $y$  nonempty. We say that  $z$  is *y-bordered* if  $z \neq y$  and  $y$  is both a prefix and a suffix of  $z$ . There are two types of *y*-bordered words: one where the prefix and suffix  $y$  do not overlap in  $z$  (that is, where  $|y| \leq |z|/2$ ), and one where they do (that is, where  $|y| > |z|/2$ ). In the first case, we say that  $z$  is *disjoint y*-bordered, and in the second case, *overlapping y*-bordered. For example, **entanglement** is disjoint **ent**-bordered, and **alfalfa** is overlapping **alfa**-bordered. For more about borders of words, see, for example, [2].

We will need two lemmas about bordered words.

**Lemma 1.** *Suppose  $z \in \Sigma^*$  is *y*-bordered. Then there exist words  $u \in \Sigma^+$  and  $v \in \Sigma^*$  and an integer  $e \geq 0$  such that  $y = (uv)^e u$  and  $z = (uv)^{e+1} u$ .*

*Proof.* Follows immediately from the Lyndon-Schützenberger theorem (see, for example, [6] or [9, Theorem 2.3.2]).  $\square$

**Lemma 2.** *Let  $u \in \Sigma^+$ ,  $v \in \Sigma^*$ , and  $e \geq 0$ . Suppose that  $y = (uv)^e u$ . Define  $z_1 = (uv)^{e+1}$  and  $z_2 = (uv)^{e+2}$ . Let  $c = |z_1|_y$  and  $d = |z_2|_y - |z_1|_y$ . Then  $c, d \geq 1$  and  $|(uv)^i|_y = (i - e)d + c - d$  for all integers  $i > e$ .*

*Proof.* If  $x = (uv)^{e+1}$ , then  $|x|_y = c$ . Appending  $uv$  to  $x$  on the right results in  $d \geq 1$  additional copies of  $y$ . The result now follows by induction.  $\square$

We also recall the following classical result.

**Theorem 1.** *Let  $x, y$  be nonempty words. There exists a word with two distinct factorizations as a concatenation of  $x$ 's and  $y$ 's if and only if  $xy = yx$ .*

*Proof.* This follows from the so-called “defect theorem” [5], or from [3, Theorem 1].  $\square$

### 3 Automata

We will need the following well-known result about pattern-matching automata (for example, see [1, §32.3]).

**Theorem 2.** *Given a word  $w \in \Sigma^n$ , a DFA  $M = (\{q_0, \dots, q_n\}, \Sigma, \delta, q_0, \{q_n\})$  exists of  $n + 1$  states such that  $\delta(q_0, x) = q_n$  if and only if  $w$  is a subword (resp., suffix) of  $x$ . Here the state  $q_i$  can be interpreted as asserting that the longest suffix of the input that matches a prefix of  $w$  is of length  $i$ .*

### 4 Interlacing

Suppose  $y$  is a subword of every  $x$ -bordered word. In this case we say  $x$  is *interlaced by*  $y$ . For example, it is easy to check that 000100 is interlaced by 1000 when the underlying alphabet  $\Sigma$  is  $\{0, 1\}$ . The following lemma gives the fundamental property of interlacing:

**Lemma 3.** *Suppose  $x$  is interlaced by  $y$ , and suppose  $z$  is a word satisfying  $|z|_y = |z|_x + k$ . Then for all  $t$  we have  $|zt|_y \geq |zt|_x + k - 1$ . In particular,  $|t|_y \geq |t|_x - 1$  for all  $t$ .*

*Proof.* Identify the starting positions of all occurrences of  $x$  in  $zt$ . Since  $x$  is interlaced by  $y$ , between any two consecutive occurrences of  $x$ , there must be at least one occurrence of  $y$ . So if  $zt$  has  $i$  more occurrences of  $x$  than  $z$  does, then  $zt$  must have at least  $i - 1$  more occurrences of  $y$  than  $z$  does.  $\square$

### 5 The language $L_{x < y}$

**Theorem 3.** *The language  $L_{x < y}$  is regular if and only if either  $x$  is interlaced by  $y$  or  $y$  is interlaced by  $x$ .*

*Proof.*  $\Leftarrow$ : There are two cases: (i)  $x$  is interlaced by  $y$ ; and (ii)  $y$  is interlaced by  $x$ .

**Case (i):** Using Lemma 3, we can build a finite automaton  $M$  recognizing  $L_{x < y}$  as follows: using the pattern-matching automata for  $x$  and  $y$  described in Section 3, on input  $z$  the machine  $M$  records whether

- (a)  $|z|_x = |z|_y + 1$ ;
- (b)  $|z|_x = |z|_y$ ;
- (c)  $|z|_x = |z|_y - 1$ , and  $|z'|_x \geq |z'|_y - 1$  for all prefixes  $z'$  of  $z$ ;
- (d)  $|z'|_x \leq |z'|_y - 2$  for some prefix  $z'$  of  $z$ .

Of course we do not maintain the actual numbers  $|z|_x$  and  $|z|_y$  in  $M$ , but only which of (a)–(d) hold. Lemma 3 implies that the four cases above cover all the possibilities. It is not possible to have  $|z|_x \geq |z|_y + 2$ , and if (d) ever occurs, we know from Lemma 3 that  $|z|_x < |z|_y$  for all words  $z$  extending  $z'$ . So in this case the correct action is for the automaton to remain in state (d), an accepting state that loops to itself on all inputs. The automaton accepts the input if and only if it is in the states corresponding to conditions (c) and (d).

**Case (ii):** Using Lemma 3, as in Case (i), we can build a finite automaton recognizing  $L_{x < y}$  as follows: using the pattern-matching automata for  $x$  and  $y$  described in Section 3, on input  $z$  the machine  $M$  records whether

- (a)  $|z|_y = |z|_x + 1$ ;
- (b)  $|z|_y = |z|_x$ , and  $|z'|_y \geq |z'|_x$  for all prefixes  $z'$  of  $z$ ;
- (c)  $|z'|_y \leq |z'|_x - 1$  for some prefix  $z'$  of  $z$ .

Lemma 3 implies that the three cases above cover all the possibilities. It is not possible to have  $|z|_y \geq |z|_x + 2$ , and if (c) ever occurs, we know from Lemma 3 that  $|z|_x \geq |z|_y$  for all words  $z$  extending  $z'$ . So in this case the correct action is for the automaton to remain in state (c), a rejecting “dead” state that loops to itself on all inputs. The automaton accepts the input if and only if it is in the state corresponding to condition (a).

$\implies$ : We proceed by proving the contrapositive. So suppose that there is some  $y$ -bordered word  $r$  such that  $x$  is not a subword of  $r$ , and some  $x$ -bordered word  $s$  such that  $y$  is not a subword of  $s$ . Using Lemma 1, we know that there are words  $u, v, p, q$  and natural numbers  $e, f$  such that  $r = (uv)^{e+1}u$ , and  $y = (uv)^e u$ , and  $s = (pq)^{f+1}p$ , and  $x = (pq)^f p$ .

Suppose that  $x$  is a subword of  $(uv)^i u$  for some  $i \geq 0$ . Since  $x$  is not a subword of  $r$ , we know that  $i \geq e + 2$ . If  $x$  is a subword of  $(uv)^{e+2}u$  and not a subword of  $(uv)^{e+1}u$ , then  $y = (uv)^e u$  must be a subword of  $x$ . But then  $y$  is a subword of  $s$ , a contradiction. So  $x$  is not a subword of  $(uv)^i u$  for any  $i$ . By exactly the same reasoning we deduce that  $y$  is not a subword of  $(pq)^j p$  for any  $j \geq 0$ .

Let  $c = |(uv)^{e+1}|_y$  and  $d = |(uv)^{e+2}|_y - |(uv)^{e+1}|_y$ . Similarly, define  $c' = |(pq)^{f+1}|_x$  and  $d' = |(pq)^{f+2}|_x - |(pq)^{f+1}|_x$ . Consider a word  $z = (uv)^i (pq)^j$ , where  $i > e$  and  $j > f$ . From above and Lemma 2, we know that  $|(uv)^i|_x = 0$  for all  $i \geq 0$  and  $|(pq)^j|_x = (j - f)d' + c' - d'$  for  $j > f$ . Let  $m$  be the number of additional occurrences of  $x$  that straddle the boundary between  $(uv)^{e+1}$  and  $(pq)^{f+1}$ . That is,  $m$  is the number of distinct values for  $k$ , such that  $x$  is a subword of  $(uv)^{e+1}(pq)^{f+1}$  starting at index  $k$  and  $(e+1)|uv| + 2 - |x| \leq k \leq (e+1)|uv| + 1$ . Similarly, we know that  $|(uv)^i|_y = (i - e)d + c - d$  for  $i > e$  and  $|(pq)^j|_y = 0$  for all  $j \geq 0$ . Let  $n$  be the number of additional occurrences of  $y$  that straddle the boundary between  $(uv)^{e+1}$  and  $(pq)^{f+1}$ . The precise definition of  $n$  is given as above by replacing  $m$  and  $x$  with  $n$  and  $y$  respectively. Thus  $z$  has  $(j - f)d' + c' - d' + m$  occurrences of  $x$  and  $(i - e)d + c - d + n$  occurrences of  $y$ .

Now assume, contrary to what we want to prove, that  $L_{x<y}$  is regular. Define  $L = L_{x<y} \cap (uv)^e(uv)^+(pq)^f(pq)^+$ . Then  $L$  is regular. Define a morphism  $h : \{a, b\}^* \rightarrow \Sigma^*$  as follows:  $h(a) = uv$ , and  $h(b) = pq$ . We claim that  $h^{-1}(z) = \{a^i b^j\}$ . One direction is clear. For the other, suppose  $h^{-1}(z)$  included some word other than  $a^i b^j$ . Then by Theorem 1, we know that  $uv$  and  $pq$  commute. But then by the Lyndon-Schützenberger theorem [6],  $uv$  and  $pq$  are both powers of some word  $t$ . But then  $x$  would be a subword of  $(uv)^\ell u$  for some  $\ell$ , which we already saw to be impossible.

By a well-known theorem (e.g., [9, Theorem 3.3.9]),  $h^{-1}(L)$  is regular. But  $h^{-1}(L) = \{a^i b^j : (i - e)d + c - d + n < (j - f)d' + c' - d' + m, \text{ for } i > e, j > f\}$  which, using the pumping lemma, is not regular.  $\square$

## 6 The language $L_{x=y}$

**Theorem 4.** *The language  $L_{x=y}$  is regular if and only if either  $x$  is interlaced by  $y$  or  $y$  is interlaced by  $x$ .*

*Proof.* The proof is quite similar to the case  $L_{x<y}$ , and we indicate only what needs to be changed.

$\Leftarrow$ : Without loss of generality we can assume that  $x$  is interlaced by  $y$ . Using Lemma 3 we can build a finite automaton recognizing  $L_{x=y}$  just as we did for  $L_{x<y}$ , using case (i). The only difference now is that the accepting state corresponds to (b).

$\Rightarrow$ : Proceeding by contraposition, suppose that there is some  $y$ -bordered word  $r$  such that  $x$  is not a subword of  $r$ , and some  $x$ -bordered word  $s$  such that  $y$  is not a subword of  $s$ . Once again, we follow the argument used for  $L_{x<y}$ , but there is one difference.

Recall that  $z = (uv)^i(pq)^j$  for some  $i > e$  and  $j > f$ . By the argument for  $L_{x<y}$  we know that  $z$  has  $(j - f)d' + c' - d' + m$  occurrences of  $x$  and  $(i - e)d + c - d + n$  occurrences of  $y$ . Let  $A = (-(m + c')) \bmod d'$  and  $B = (-(n + c)) \bmod d$ . Let  $w$  be the shortest suffix of  $(uv)^{e+2}$  such that  $wz$  has  $(i - e)d + c - d + n + B$  occurrences of  $y$ ; let  $w'$  be the shortest prefix of  $(pq)^{f+2}$  such that  $zw'$  has  $(j - f)d' + c' - d' + m + A$  occurrences of  $x$ . Then by our construction  $wzw'$  has  $(j - f + C)d'$  occurrences of  $x$  and  $(i - e + D)d$  occurrences of  $y$ , for some  $C, D \geq 0$ .

Now assume, contrary to what we want to prove, that  $L_{x=y}$  is regular. Define  $L' = L_{x=y} \cap w(uv)^e(uv)^+(pq)^f(pq)^+w'$ . Then  $L'$  is regular. Define  $L = \#L'\#$ , where  $\#$  is a new symbol not in the alphabet  $\Sigma$ ; then  $L$  is regular. Define a morphism  $h : \{a, b, a', b'\}^* \rightarrow \Sigma^*$  as follows:  $h(a') = \#w$ ,  $h(a) = uv$ ,  $h(b) = pq$ , and  $h(b') = w'\#$ . We claim that  $h^{-1}(\#wzw'\#) = \{a' a^i b^j b'\}$ . One direction is clear, and the other follows from Theorem 1. By a well-known theorem (e.g., [9, Theorem 3.3.9]),  $h^{-1}(L)$  is regular. But  $h^{-1}(L) = \{a' a^i b^j b' : (i - e + D)d = (j - f + C)d', \text{ for } i > e, j > f\}$  which, using the pumping lemma, is not regular.  $\square$

## 7 Testing the criteria

Given  $x, y$  we can test if there is some  $y$ -bordered word  $z$  such that  $x$  is not a subword of  $z$ , as follows: create a DFA recognizing the language

$$(\Sigma^* x \Sigma^*)^c \cap y \Sigma^+ \cap \Sigma^+ y.$$

A simple construction gives such a DFA  $M$  with at most  $N = (|x| + 1)(|y| + 3)(|y| + 2)$  states and at most  $N|\Sigma|$  transitions.

This can be improved to  $N' = (|x| + 1)(2|y| + 3)$  states as follows: first build a DFA of  $(2|y| + 3)$  states recognizing the language  $y \Sigma^+ \cap \Sigma^+ y$  by “grafting” the DFA,  $A_1$ , of  $|y| + 3$  states recognizing  $y \Sigma^+$  onto the DFA,  $A_2$ , of  $|y| + 2$  states recognizing  $\Sigma^+ y$ . This can be done by modifying the pattern-matching DFA described in Theorem 2. Simply replace transitions to the final state in  $A_1$  with transitions to the appropriate states in  $A_2$ . The final state of  $A_1$  and the initial state of  $A_2$  both become unreachable. Then form the direct product with the DFA for  $(\Sigma^* x \Sigma^*)^c$ . The resulting DFA has  $N'$  states. We can then use a depth-first search on the underlying transition graph of  $M$  to check if  $L(M) \neq \emptyset$ .

Thus, we have proved:

**Corollary 1.** *There is an algorithm running in time  $O(|\Sigma||x||y|)$  that decides whether the criteria of Theorems 3 and 4 hold.*

**Corollary 2.** *If there exists a  $y$ -bordered word  $z$  such that  $x$  is not a subword of  $z$ , then  $|z| < N'$ .*

*Proof.* If  $M = (Q, \Sigma, \delta, q_0, F)$  accepts any word at all, then it accepts a word of length at most  $|Q| - 1$ .  $\square$

## 8 Improving the bound in Corollary 2

As we have seen in Corollary 2, if  $x$  is not a subword of some  $y$ -bordered word, then there is a relatively short “witness” to this fact. We now show that this witness can be taken to be of the form  $yty$  for some  $t$  of *constant length*. The precise constant depends on the cardinality of the underlying alphabet  $\Sigma$ . In Corollary 3 we prove that if  $|\Sigma| \geq 3$ , then this constant is 1. In Corollary 4 we prove that if  $|\Sigma| = 2$ , then this constant is 3.

**Theorem 5.** *Suppose  $\Sigma$  is an alphabet that contains at least three symbols, and let  $x, y \in \Sigma^*$ . Without loss of generality assume that  $\{0, 1, 2\} \subseteq \Sigma$ . If  $x$  is a subword of  $y0y$  and  $y1y$  and  $y2y$ , then  $x$  is a subword of  $y$ .*

*Proof.* Assume, contrary to what we want to prove, that  $x$  is not a subword of  $y$ . Also assume that  $|y| = m$  and  $|x| = n$ . For  $x$  to be a subword of  $y0y$  (resp.,  $y1y$ ,  $y2y$ ), then, it must be that  $x$  “straddles” the  $y$ — $y$  boundary. More precisely, when we consider where  $x$  appears inside  $y0y$ , the first symbol of  $x$  must occur at or to the left of position  $m + 1$  of  $y0y$

(resp.,  $y1y$ ,  $y2y$ ). Similarly, the last symbol of  $x$  must occur at or to the right of position  $m + 1$  of  $y0y$  (resp.,  $y1y$ ,  $y2y$ ).

For  $a = 0, 1, 2$ , label the  $x$  that matches  $yay$  as  $x_a$ , and assume that the position of the 0 that matches  $x_0$  is  $i$ , the position of the 1 that matches  $x_1$  is  $j$ , and the position of the 2 that matches  $x_2$  is  $k$ . Note that  $x_0 = x_1 = x_2 = x$ ; the indices just allow us to refer to the diagram below. Without loss of generality we can assume  $1 \leq i < j < k \leq n$ . Thus we obtain a picture as in Figure 2. Here we have labeled the two occurrences of  $y$  as  $y$  and  $y'$ , so we can refer to them unambiguously. Note that  $i \geq 1$  and  $k \leq m + 1$ . Furthermore, note that  $n \leq m + i$ .

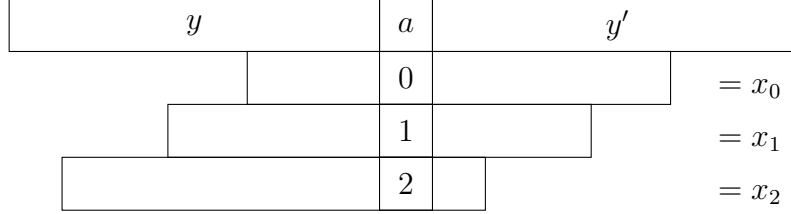


Figure 2: Matches of  $x$  against  $y0y$ ,  $y1y$ , and  $y2y$

We now use “index-chasing” to show that  $x[k] = x[i]$ ; this will give us a contradiction, since  $x[k] = 2$  and  $x[i] = 0$ . We will use the following identities, which can be deduced by observing Figure 2.

$$x_1[\ell] = y[\ell + m + 1 - j] \text{ for } 1 \leq \ell \leq j - 1; \quad (1)$$

$$x_2[\ell] = y[\ell + m + 1 - k] \text{ for } 1 \leq \ell \leq k - 1; \quad (2)$$

$$x_0[\ell] = y'[\ell - i] \text{ for } i + 1 \leq \ell \leq n; \quad (3)$$

$$x_1[\ell] = y'[\ell - j] \text{ for } j + 1 \leq \ell \leq n. \quad (4)$$

Notice that  $j + 1 \leq k \leq n$ , so we can take  $\ell = k$  in (4) to get  $x[k] = y[k - j]$ . Additionally,  $k - j \geq 1$ , giving  $i + 1 \leq i + k - j$ . Also  $i - j < 0 \leq n - k$ , so  $i + k - j \leq n$ . Thus we can take  $\ell = i + k - j$  in (3) to obtain  $y[k - j] = x[i + k - j]$ .

Since  $i \geq 1$  and  $k - j \geq 1$ , we get  $i + k - j \geq 2$ . Since  $j - i \geq 1$  we have  $i - j \leq -1$  and  $i + k - j \leq k - 1$ . Thus we can take  $\ell = i + k - j$  in (2) to get  $x[i + k - j] = y[i + m + 1 - j]$ . Since  $1 \leq i \leq j - 1$ , we can take  $\ell = i$  in (1) to get  $y[i + m + 1 - j] = x[i]$ . Putting these observations together, we finally obtain

$$2 = x[k] = y[k - j] = x[i + k - j] = y[i + m + 1 - j] = x[i] = 0,$$

which produces the desired contradiction. □

**Corollary 3.** *Suppose  $|\Sigma| \geq 3$ . Then  $x$  is a subword of  $yty$  for all  $t$  with  $|t| = 1$  if and only if  $y$  is interlaced by  $x$ .*



We now turn to case of a binary alphabet. This case is more subtle. For example, consider when  $x = 10100$  and  $y = 01001010$ . Then, as can be verified,  $x$  is a subword of the self-overlaps  $y(010)^{-1}y$  and  $y0^{-1}y$ , as well as the words  $yy, y0y, y1y, y00y, y01y, y10y, y11y, y000y, y001y, y010y, y011y, y100y, y101y$ . But  $x$  is not a subword of  $y110y$ .

For a binary alphabet  $\Sigma$ , a special role is played by the language

$$A = 01^+ \cup 10^+ \cup 0^+1 \cup 1^+0.$$

We also define the following languages. For each integer  $k \geq 1$ , let  $B_{0^k1} := (1 + 01 + \dots + 0^{k-1}1)^+0^k0^*$  and  $B_{10^k} := 0^*0^k(1 + 10 + \dots + 10^{k-1})^+$ . Similarly, define  $B_{1^k0}$  and  $B_{01^k}$  by relabeling 0 to 1 and 1 to 0.

**Lemma 4.** *Suppose  $\Sigma = \{0, 1\}$  and  $x \in A$ . Then  $y \in B_x$  if and only if  $x$  is not a subword of  $y$ , but  $x$  is a subword of all  $y$ -bordered words.*

*Proof.* We consider the case where  $x = 0^k1$  and note that the case where  $x = 1^k0$  is given by relabeling 0 to 1 and 1 to 0, and the other two cases are given by a symmetric argument.

$\Rightarrow$ : Suppose that  $y \in B_x = B_{0^k1} = (1 + 01 + \dots + 0^{k-1}1)^+0^k0^*$  and  $z$  is a  $y$ -bordered word. By the definition of  $B_x$ , observe that  $x$  is not a subword of  $y$ . By Lemma 1 there exist  $u \in \Sigma^+$ , and  $v \in \Sigma^*$ , and a natural number  $e \geq 0$  such that  $y = (uv)^e u$  and  $z = (uv)^{e+1}u$ .

We first show that  $e \leq 1$ . If we assume the contrary, then  $y = (uv)^{e-2}uvuvu$ . We know that  $vu$  has the suffix  $10^{k+i}$  for some  $i \geq 0$ . But since there is at least one 1 in  $vu$  we have that  $0^k1$  is a subword of  $uvuvu$ , giving a contradiction.

If  $e = 0$  then  $z = uvu = yvy$  for some  $v \in \Sigma^*$ . Since  $vy$  has at least one 1 and  $y$  has a suffix of  $0^k$ , we get that  $x$  is a subword of  $yvy = z$ . If  $e = 1$  then  $y = uvu$  such that  $vu$  has the suffix  $0^k$  and there is at least one 1 in  $vu$ . Then  $z = (uv)^2u = uvuvu$  has  $0^k1$  as a subword.

$\Leftarrow$ : Assume, to get a contradiction, that there is some  $y \in \Sigma^* \setminus B_x = \Sigma^* \setminus B_{0^k1}$  such that  $x$  is a subword of all  $y$ -bordered words and  $x$  is not a subword of  $y$ . Then  $y$  satisfies at least one of the following cases, and we will get a contradiction in each of these.

**Case (i):**  $y = 0^i$  for some  $i \geq 0$ . Clearly,  $x$  is not a subword of the  $y$ -bordered word  $yy$ .

**Case (ii):**  $y$  has  $x = 0^k1$  as a subword, giving an immediate contradiction.

**Case (iii):** The suffix of  $y$  is  $10^i$  for some  $0 \leq i < k$ . Then consider the  $y$ -bordered word  $z = y1y$ . If  $x$  is a subword of  $z$  but  $x$  is not a subword of  $y$ , then  $x$  must straddle the  $y$ — $y$  boundary in  $z$ . So the 1 in  $x = 0^k1$  must align with the 1 between the  $y$ 's in  $z = y1y$ . But the suffix of  $y$  is  $10^i$  for  $i < k$ . So  $x$  cannot be a subword of  $y1y$ .  $\square$

However, for  $x \notin A$ , it turns out that if  $x$  is not a subword of  $y$ , then there is some word  $t$  of length 3 such that  $x$  is not a subword of  $yty$ . To prove this we first give two preliminary lemmas.

**Lemma 5.** *Suppose  $\Sigma = \{0, 1\}$ , and let  $x, y \in \Sigma^*$  with  $|x| = n$  and  $|y| = m$ . Suppose  $x$  is not a subword of  $y$ , but  $x$  is a subword of  $yty$  for all  $t \in \Sigma^*$  such that  $|t| = 3$  and  $x \notin A$ . Then for every integer  $k$  satisfying  $\max\{1, m - n + 2\} \leq k \leq \min\{2m + 3 - n, m + 2\}$  and*

for all pairs of words  $t_1, t_2 \in \Sigma^*$  with  $|t_1| = |t_2| = 3$ , we have either  $x \neq (yt_1y)[k..k+n-1]$  or  $x \neq (yt_2y)[k+1..k+n]$ , or both.

*Proof.* Assume, to get a contradiction, that there exist  $x, y \in \Sigma^*$  such that  $x$  is not a subword of  $y$  and  $x \notin A$  and that there exist  $t_1, t_2 \in \Sigma^*$  with  $|t_1| = |t_2| = 3$  and an integer  $k$  satisfying  $\max\{1, m-n+2\} \leq k \leq \min\{2m+3-n, m+2\}$  such that  $(yt_1y)[k..k+n-1] = (yt_2y)[k+1..k+n] = x$ , and furthermore  $x$  is a subword of  $yty$  for all  $t \in \Sigma^*$  with  $|t| = 3$ . Let  $t_1 = a_1b_1c_1$  and  $t_2 = a_2b_2c_2$  and  $x = x_1x_2 \cdots x_n$ . Before proceeding, first observe that  $n \geq 3$  since for all  $x$  with  $|x| \leq 2$  we have that either  $x \in A$  or  $x$  is not a subword of one of  $y000y$  and  $y111y$ .

**Case (i):**  $\max\{1, m-n+3\} \leq k \leq \min\{2m+3-n, m+1\}$ .

If  $\max\{1, m-n+4\} \leq k \leq \min\{2m+3-n, m\}$  then  $n \geq 4$  and we can write  $x = x_1va_1b_1c_1w = va_2b_2c_2wx_n$  for some  $v, w \in \Sigma^*$  where  $x_1v = va_2$  and  $c_1w = wx_n$  and  $a_1b_1 = b_2c_2$ . Then, by the first theorem of Lyndon-Schützenberger, we have that  $v = x_1^i$  and  $w = x_n^j$  for integers  $i, j \geq 0$ . Thus  $x$  can be re-written as  $x = x_1^i a_1 b_1 x_n^j$  for  $x_1, a_1, b_1, x_n \in \Sigma$  and  $i, j \geq 1$ .

If  $k = m - n + 3$  then, where  $n \geq 3$ , we can write  $x = x_1va_1b_1 = va_2b_2c_2$  for  $v \in \Sigma^*$  and after applying the first theorem of Lyndon-Schützenberger we get  $x = x_1^i a_1 b_1$  where  $i \geq 1$ .

Similarly if  $k = m + 1$  then we can write  $x = a_1b_1c_1w = b_2c_2wx_n$  and applying the first theorem of Lyndon-Schützenberger gives  $x = a_1b_1x_n^j$  for  $j \geq 1$ .

So we have that  $x = x_1^i a_1 b_1 x_n^j$  for  $a_1, b_1, x_1, x_n \in \Sigma$  and  $i, j \geq 0$  and either  $i \geq 1$  or  $j \geq 1$ . We will proceed by getting a contradiction for each possible assignment of  $a_1, b_1, x_1, x_n$  to symbols in  $\Sigma$  for all valid  $i, j$ . Table 1 gives contradictions for all possible assignments where  $x_1 = 0$ . Note that the remaining cases can be ruled out by relabeling 0 to 1 and 1 to 0.

**Case (iii):**  $k = m - n + 2 \geq 1$ .

We can write  $x = x_1wa_1 = wa_2b_2$  for some  $w \in \Sigma^+$ , where  $x_1w = wa_2$ . So by the first theorem of Lyndon-Schützenberger, we get  $w = x_1^i$  for some integer  $i \geq 1$  and thus  $x = x_1^{i+1}x_n$  for  $x_1, x_n \in \Sigma$ . If  $x_1 = x_n$ , then  $x_1^{i+1}x_n = x_1^{i+2}$ . But, since  $x$  is not a subword of  $y$  we cannot have that  $x_1^{i+2}$  is a subword of both  $y111y$  and  $y000y$ , giving a contradiction. If instead we have  $x_1 \neq x_n$ , then  $x_1^{i+1}x_n \in A$ , an immediate contradiction.

**Case (iv):**  $k = m + 2 \leq 2m + 3 - n$ .

We can write  $x = b_1c_1w = c_2wx_n$  and similar to Case (iii), we get  $x = x_1x_n^{i+1}$  for  $x_1, x_n \in \Sigma$  and  $i \geq 1$ . By the same argument as in Case (iii), we get a contradiction if  $x_1 = x_n$  and if  $x_1 \neq x_n$ .

Table 1: Contradictions for each  $a_1, b_1, x_n \in \Sigma$  and  $x_1 = 0$ , where in each row  $x = x_1^i a_1 b_1 x_n^j$  and  $i, j \geq 0$  and either  $i \geq 1$  or  $j \geq 1$ . The contradictions rely on the assumption that  $x$  is not a subword of  $y$ .

| $x_1$ | $a_1$ | $b_1$ | $x_n$ | Contradiction  |
|-------|-------|-------|-------|--|
| 0     | 0     | 0     | 0     | For all $i, j \geq 0$ we have that $x$ is not a subword of $y111y$ .   |
| 0     | 0     | 0     | 1     | For all $i \geq 0$ :<br>If $j = 0$ , then $x$ is not a subword of $y111y$ ;<br>If $j = 1$ , then $x = 0^{i+2}1 \in A$ ;<br>If $j > 1$ , then if $x$ is a subword of $y101y$ , then $y$ has $0^{i+2}1^{j-1}$ as a suffix. But if $x$ is a subword of $y011y$ , then $y$ has $0^{i+1}$ as a suffix.  |
| 0     | 0     | 1     | 0     | For all $i \geq 0$ :<br>If $j = 0$ , then $x = 0^{i+1}1 \in A$ ;<br>If $j > 0$ , then $x$ is not a subword of $y111y$ .  |
| 0     | 0     | 1     | 1     | If $i = 0$ or $j = 0$ , then $x \in A$ .<br>If $i > 0$ and $j > 0$ , then if $x$ is a subword of $y101y$ , then $y$ has $0^{i+1}1^j$ as a suffix. But if $x$ is a subword of $y011y$ , then $y$ has $0^i$ as a suffix.   |
| 0     | 1     | 0     | 0     | If $i = 0$ , then $x = 10^{j+1} \in A$ .<br>If $i > 0$ , then $x$ is not a subword of $y111y$ .  |
| 0     | 1     | 0     | 1     | If $i = 0$ and $j > 0$ , then $x$ is not a subword of $y000y$ .<br>If $i > 0$ and $j = 0$ , then $x$ is not a subword of $y111y$ .<br>If $i > 0$ and $j = 1$ , then if $x$ is a subword of $y011y$ , then $y$ has $0^i1$ as a suffix. But if $x$ is a subword of $y111y$ , then $y$ has $0^i10$ as a suffix.<br>If $i = 1$ and $j > 0$ , then if $x$ is a subword of $y001y$ , then $y$ has $01^j$ as a prefix. But if $x$ is a subword of $y000y$ , then $y$ has $101^j$ as a prefix.<br>If $i > 1$ and $j > 1$ , then if $x$ is a subword of $y011y$ , then $y$ has $0^i1$ as a suffix. But if $x$ is a subword of $y111y$ , then $y$ has $0^i101^\ell$ as a suffix for some $\ell < j$ . Since $i > 1$ , this is a contradiction. |
| 0     | 1     | 1     | 0     | If $i = 0$ and $j = 1$ , then $x = 110 \in A$ .<br>If $i = 1$ and $j = 0$ , then $x = 011 \in A$ .<br>If $i > 1$ and $j = 0$ , then if $x$ is a subword of $y101y$ , then $y$ has $0^i1$ as a suffix. But if $x$ is a subword of $y011y$ , then $y$ has $0^{i-1}$ as a suffix.<br>If $i = 0$ and $j > 1$ , then if $x$ is a subword of $y101y$ , then $y$ has $10^j$ as a prefix. But if $x$ is a subword of $y110y$ , then $y$ has $0^{j-1}$ as a prefix.<br>If $i > 0$ and $j > 0$ , then $x$ is not a subword of $y111y$ .  |
| 0     | 1     | 1     | 1     | For all $j \geq 0$ :<br>If $i = 0$ , then $x$ is not a subword of $y000y$ ;<br>If $i = 1$ , then $x = 01^{j+2} \in A$ ;<br>If $i > 1$ , then if $x$ is a subword of $y011y$ , then $y$ has $0^{i-1}$ as a suffix.<br>But if $x$ is a subword of $y101y$ , then $y$ has $0^i1^{j+1}$ as a suffix.   |

□

**Lemma 6.** Suppose  $\Sigma = \{0, 1\}$ , and let  $x, y \in \Sigma^*$  with  $|x| = n$  and  $|y| = m$ . If  $x$  is a subword of  $yty$  for all  $t \in \Sigma^*$  such that  $|t| = 3$  and  $x \notin A$ , and  $x$  is not a subword of  $y$ , then for all pairs of words  $t_1, t_2 \in \Sigma^*$  with  $|t_1| = |t_2| = 3$  we have either  $x \neq (yt_1y)[m+1..m+n]$ , or  $x \neq (yt_2y)[m+3..m+2+n]$ , or both.

*Proof.* Assume, to get a contradiction, that there exist  $x, y \in \Sigma^*$  such that  $x$  is not a subword of  $y$  and  $x \notin A$  and that there exist  $t_1, t_2 \in \Sigma^*$  with  $|t_1| = |t_2| = 3$  such that  $(yt_1y)[m+1..m+n] = (yt_2y)[m+3..m+2+n] = x$ , and furthermore  $x$  is a subword of  $yty$  for all  $t \in \Sigma^*$  with  $|t| = 3$ . Let  $t_1 = a_1b_1c_1$  and  $t_2 = a_2b_2c_2$  and  $x = x_1x_2 \cdots x_n$ , and assume  $|y| = m$ .

We can write  $x = a_1b_1c_1w = c_2wx_{n-1}x_n$ . So  $b_1c_1w = wx_{n-1}x_n$  and by the first theorem of Lyndon-Schützenberger there exist  $u \in \Sigma^+$  and  $v \in \Sigma^*$  and an integer  $i \geq 0$  such that  $b_1c_1 = uv$  and  $x_{n-1}x_n = vu$  and  $w = (uv)^i u = u(vu)^i$ . This gives  $x = x_1wx_{n-1}x_n = x_1(uv)^i uvu = x_1(uv)^{i+1}u$ . We now consider each possible  $u \in \Sigma^+$  and  $v \in \Sigma^*$ , seeking a contradiction in each case. The contradictions are summarized in Table 2. Note again that the contradictions are given for all cases where  $x_1 = 0$ ; the remaining cases can be obtained by relabeling 0 to 1 and 1 to 0.

Table 2: Contradictions for each valid  $u \in \Sigma^+$ ,  $v \in \Sigma^*$ , and  $x_1 = 0$ , where in each row  $x = x_1(uv)^{i+1}u$  for  $i \geq 0$ . The contradictions rely on the assumption that  $x$  is not a subword of  $y$ .

| $x_1$ | $u$ | $v$        | Contradiction   |
|-------|-----|------------|---|
| 0     | 00  | $\epsilon$ | $x$ is not a subword of $y111y$ .   |
| 0     | 0   | 0          | $x$ is not a subword of $y111y$ .   |
| 0     | 01  | $\epsilon$ | If $x$ is a subword of $y111y$ , then $y$ has $0(01)^{i+1}0$ as a suffix.<br>But if $x$ is a subword of $y011y$ , then $y$ has $0(01)^{i+1}$ as a suffix. |
| 0     | 0   | 1          | $x$ is not a subword of $y111y$ .   |
| 0     | 10  | $\epsilon$ | $x$ is not a subword of $y111y$ .   |
| 0     | 1   | 0          | If $x$ is a subword of $y111y$ then $y$ has $0(10)^{i+1}$ as a suffix.<br>But if $x$ is a subword of $y011y$ then $y$ has $(01)^{i+1}$ as a suffix.       |
| 0     | 11  | $\epsilon$ | $x \in A$ .   |
| 0     | 1   | 1          | $x \in A$ .   |

□

**Theorem 6.** Suppose  $\Sigma = \{0, 1\}$  and let  $x, y \in \Sigma^*$ . If  $x$  is a subword of  $yty$  for all  $t \in \Sigma^*$  such that  $|t| = 3$  and  $x \notin A$ , then  $x$  is a subword of  $y$ .

*Proof.* Define the function  $f : \Sigma^* \times \Sigma^* \rightarrow \mathbb{N}$  over pairs of words  $x, w \in \Sigma^*$  such that  $x$  is a subword of  $w$  as  $f(x, w) = \min\{i \in \mathbb{N} : w[i..i + |x| - 1] = x\}$ . Also, define the bitwise complements of elements of  $\Sigma$  as  $\bar{0} = 1$  and  $\bar{1} = 0$ .

Assume, to get a contradiction, that  $x$  is not a subword of  $y$  and also assume that  $|y| = m$  and  $|x| = n$ . If  $x$  is a subword of  $yty$  for each  $t \in \Sigma^*$  with  $|t| = 3$ , then for each such  $t$  we have  $f(x, yty) \leq m + 3$  and  $f(x, yty) + n - 1 \geq m + 1$ . Since the position of  $x$  in  $yty$  cannot be the same for all valid  $t$ , let  $t_0 = a_0b_0c_0$  be the choice of  $t$  for which  $f(x, yty)$  is greatest across all valid  $t$  and let  $t_4 = a_4b_4c_4 \neq t_0$  be the choice of  $t$  for which  $f(x, yty)$  is smallest across all valid  $t$ . We now consider two cases depending on the position of  $x$  in  $yt_0y$ .

**Case (i):**  $f(x, yt_0y) = m + 3$ . Consider  $t_1 = \bar{a}_40\bar{c}_0$  and  $t_2 = \bar{a}_41\bar{c}_0$ . Since  $t_1$  and  $t_2$  differ from  $t_4$  in the first index of  $t_4$ , and  $t_4$  gives the leftmost position for  $x$  as a subword of  $yty$  over all valid choices of  $t$ , we know  $f(x, yt_1y) \neq f(x, yt_4y)$  and  $f(x, yt_2y) \neq f(x, yt_4y)$ . Similarly we have  $f(x, yt_1y) \neq f(x, yt_0y)$  and  $f(x, yt_2y) \neq f(x, yt_0y)$ . Applying Lemma 5 to the pairs  $t_4, t_1$  and  $t_4, t_2$  and Lemmas 5 and 6 to the pairs  $t_1, t_0$  and  $t_2, t_0$  we have that  $f(x, yt_1y) + n - 1 \geq m + 3$  and  $f(x, yt_2y) + n - 1 \geq m + 3$  and that  $f(x, yt_1y) \leq m + 1$  and  $f(x, yt_2y) \leq m + 1$ . So for  $yt_1y$  (resp.,  $yt_2y$ ), the position of  $x$  is such that it entirely overlaps  $t_1$  (resp.,  $t_2$ ). But since  $t_1 \neq t_2$  we know that the positions of  $x$  as a subword of  $yt_1y$  and  $yt_2y$  are distinct, i.e.,  $f(x, yt_1y) \neq f(x, yt_2y)$ .

So suppose without loss of generality that  $f(x, yt_1y) > f(x, yt_2y)$ . We now perform an index chasing argument, similar to that of the ternary case, using  $t_0, t_1, t_2$  and seeking the contradiction  $c_0 = (yt_0y)[m+3] = (yt_2y)[m+3] = \bar{c}_0$ . We use the same labeling scheme as in the ternary case. So define  $i, j, k$  such that  $i = m+4-f(x, yt_0y)$  and  $j = m+4-f(x, yt_1y)$  and  $k = m+4-f(x, yt_2y)$ , giving  $x_0[i] = t_0[3] = c_0$  and  $x_1[j] = t_1[3] = \bar{c}_0$  and  $x_2[k] = t_2[3] = \bar{c}_0$ . Note that in this case we have  $i = 1$  by assumption and  $j \geq i + 3$  by Lemmas 5 and 6. From Figure 3 we obtain the following identities.

$$x_1[\ell] = y[\ell + m + 3 - j] \text{ for } 1 \leq \ell \leq j - 3; \quad (5)$$

$$x_2[\ell] = y[\ell + m + 3 - k] \text{ for } 1 \leq \ell \leq k - 3; \quad (6)$$

$$x_0[\ell] = y'[\ell - i] \text{ for } i + 1 \leq \ell \leq n; \quad (7)$$

$$x_1[\ell] = y'[\ell - j] \text{ for } j + 1 \leq \ell \leq n. \quad (8)$$

Since  $j + 1 \leq k \leq n$ , we can apply (8) to get  $x_1[k] = y'[k - j]$ . Then, since  $i \leq j$  and  $k \leq n$ , we have  $i + k \leq n + j$ , giving  $i + k - j \leq n$ . This, together with the inequality  $k - j \geq 1$  giving  $i + 1 \leq i + k - j$ , means that we can apply (7) to get  $y'[k - j] = x_0[i + k - j]$ . Next,  $j - i \geq 3$  gives  $i + k - j \leq k - 3$ , so applying (6) gives  $x_2[k + i - j] = y[i + m + 3 - j]$ . Finally, since  $1 \leq i \leq j - 3$ , we can apply (5) to get  $y[i + m + 3 - j] = x_1[i]$ . Together this gives the contradiction

$$\bar{c}_0 = x_2[k] = y'[k - j] = x_0[i + k - j] = y[i + m + 3 - j] = x_1[i] = c_0.$$

| $y$ | $a$              | $b$ | $c$              | $y'$ |         |
|-----|------------------|-----|------------------|------|---------|
|     |                  |     | $c_0$            |      | $= x_0$ |
|     | $\overline{a_4}$ | 0   | $\overline{c_0}$ |      | $= x_1$ |
|     | $\overline{a_4}$ | 1   | $\overline{c_0}$ |      | $= x_2$ |

Figure 3: Positions of  $x$  in  $yt_0y, yt_1y, yt_2y$  for Case (i).

**Case (ii):**  $f(x, yt_0y) \leq m + 2$ . Consider  $t_1 = \overline{a_4b_0c_0}$  and  $t_2 = \overline{a_4b_0\overline{c_0}}$  and  $t_3 = \overline{a_4b_0\overline{c_0}}$ . By the same argument as in Case 1 we get that the positions of  $x$  as a subword of each of  $yt_0y, yt_1y, yt_2y, yt_3y, yt_4y$  are all distinct. Furthermore, by Lemma 5 we have that for each pair  $t_i, t_j$  with  $0 \leq i, j \leq 4$  and  $i \neq j$  the difference in positions of  $x$  as a subword of  $yt_iy$  and  $yt_jy$  is  $|f(x, yt_iy) - f(x, yt_jy)| \geq 2$ . We now order these choices of  $t$  and relabel  $t_1, t_2, t_3$  if necessary such that  $f(x, yt_0y) > f(x, yt_3y) > f(x, yt_1y) > f(x, yt_2y) > f(x, yt_4y)$ . At this point we again perform an index-chasing argument using  $t_0, t_1, t_2$ . If we have that  $t_2[3] = \overline{c_0}$  then the argument given in Case (i) holds to give a contradiction. If instead we have that  $t_2[3] = c_0$ , then we know that  $t_2[2] = \overline{b_0}$  and we will get the contradiction  $b_0 = (yt_0y)[m+2] = (yt_2y)[m+2] = \overline{b_0}$ . To do this we define  $i, j, k$  such that  $i = m + 3 - f(x, yt_0y)$  and  $j = m + 3 - f(x, yt_1y)$  and  $k = m + 3 - f(x, yt_2y)$ , giving  $x_0[i] = t_0[2] = b_0$  and  $x_1[j] = t_1[2]$  and  $x_2[k] = t_2[2] = \overline{b_0}$ . Since  $f(x, yt_0y) > f(x, yt_3y) > f(x, yt_1y)$ , Lemma 5 gives  $f(x, yt_0y) \geq f(x, yt_3y) + 2$  and  $f(x, yt_3y) \geq f(x, yt_1y) + 2$ . So  $f(x, yt_0y) \geq f(x, yt_1y) + 4$  and thus  $j \geq i + 4$ . From Figure 4 we get the following identities. Note that these identities are centered around  $t[2]$  instead of  $t[3]$  as in Case 1.

$$x_1[\ell] = y[\ell + m + 2 - j] \text{ for } 1 \leq \ell \leq j - 2; \quad (9)$$

$$x_2[\ell] = y[\ell + m + 2 - k] \text{ for } 1 \leq \ell \leq k - 2; \quad (10)$$

$$x_0[\ell] = y'[\ell - i - 1] \text{ for } i + 2 \leq \ell \leq n; \quad (11)$$

$$x_1[\ell] = y'[\ell - j - 1] \text{ for } j + 2 \leq \ell \leq n. \quad (12)$$

Since  $j + 2 \leq k \leq n$  we can apply (12) to get  $x_1[k] = y'[k - j - 1]$ . Then since  $i \leq j$  and  $k \leq n$  we have  $i + k \leq n + j$ , giving  $i + k - j - 1 < i + k - j \leq n$ . This, together with  $k - j \geq 2$  giving  $i + 2 \leq i + k - j$  by Lemma 5, means that we can apply (11) to get  $y'[k - j - 1] = x_0[i + k - j]$ . Next,  $j - i \geq 4$  gives  $i + k - j \leq k - 4 < k - 2$ , so applying (10) gives  $x_2[k + i - j] = y[i + m + 2 - j]$ . Finally, since  $1 \leq i \leq j - 4 < j - 2$ , we can apply (9) to get  $y[i + m + 2 - j] = x_1[i]$ . Together this gives the contradiction

$$\overline{b_0} = x_2[k] = y'[k - j - 1] = x_0[i + k - j] = y[i + m + 2 - j] = x_2[i] = b_0.$$

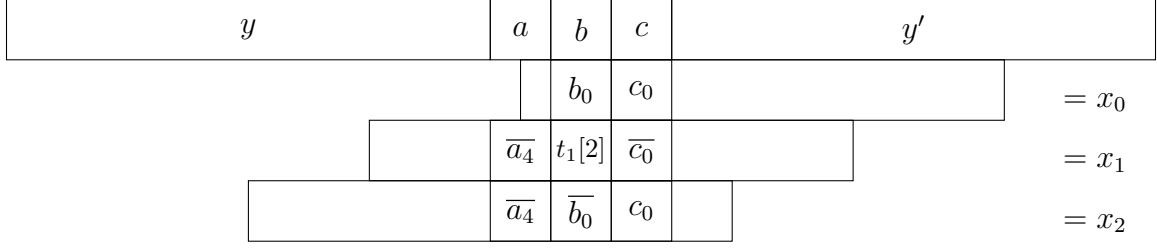


Figure 4: Positions of  $x$  in  $yt_0y, yt_1y, yt_2y$  for Case (ii) where  $t_2[3] = c_0$ .

□

**Corollary 4.** *Let  $\Sigma = \{0, 1\}$ . Then  $x$  is a subword of every  $y$ -bordered word if and only if  $x$  is a subword of  $yty$  for all words  $t$  of length 3.*

*Proof.* If  $x$  is a subword of every  $y$ -bordered word, then clearly  $x$  is a subword of  $yty$  for all words  $t$  of length 3. For the other direction there are two cases.

**Case 1:**  $x \notin A$ . Then by Theorem 6 we know  $x$  is a subword of  $y$ . So  $x$  is also a subword of every  $y$ -bordered word.

**Case 2:**  $x \in A$ . Then  $x$  has the form  $01^i$ , or  $0^i1$ , or  $10^i$ , or  $1^i0$  for some  $i \geq 1$ . We consider the case where  $x = 01^i$  and note that the case where  $x = 1^i0$  follows by a symmetric argument and the other cases are given by relabeling 0 to 1 and 1 to 0. If  $x$  is a subword of  $y$  then the result follows trivially. So suppose that  $x = 01^i$  is not a subword of  $y$ , but that  $x$  is a subword of  $yty$  for all words  $t$  of length 3. Then, since  $x$  is a subword of  $y000y$ , we have that  $1^i$  is a prefix of  $y$ . Additionally, since  $x$  is a subword of  $y111y$ , we know that  $01^j$  is a suffix of  $y$  for some  $j$  satisfying  $0 \leq j < i$ . So we have  $y = 1^i w 01^j$  for some  $w \in \Sigma^*$ . Now consider a  $y$ -bordered word  $z$ . Let  $k$  be the index of the first 0 in  $z$ . Since  $z$  has  $y$  as a prefix and a suffix, and  $z \neq y$ , we know that  $|z| \geq |y| + k$ . This is because the  $y$ -suffix of  $z$  must start after the first 0 in  $z$ . So we have that there are  $i$  consecutive 1's in  $z$  starting at some index  $\ell > k$ . Let  $k'$  be the largest index less than  $\ell$  such that  $z[k'] = 0$ . Then  $z[k'..k' + i] = 01^i$ . So  $x$  is a subword of  $z$ . □

*Remark 1.* The number 3 is optimal in Corollary 4. Consider  $x = 10100$ ,  $y = 01001010$ . Then  $x$  is a subword of every  $y$ -bordered word of length  $\leq 2|y| + 2 = 18$ , but not a subword of  $yty$  with  $t = 110$ .

## 9 Finiteness

We now examine when  $L_{x=y}$  is finite.

**Theorem 7.** *Let  $x, y \in \Sigma^+$ . Then  $L_{x=y}$  is finite if and only if  $|\Sigma| = 1$  and  $x \neq y$ .*

*Proof.* There are four cases to consider.

**Case (i):**  $x = a^i$  and  $y = a^j$  for integers  $i, j > 0$ . If  $|\Sigma| = 1$ , then  $L_{x=y}$  is finite if and only if  $x \neq y$ , for otherwise without loss of generality  $i < j$ , and for  $n \geq j$  the word  $a^n$  contains  $n - j + 1$  occurrences of  $a^j$ , but  $n - i + 1$  occurrences of  $a^i$ .

Otherwise  $|\Sigma| > 1$ . Let  $b \in \Sigma$  and  $b \neq a$ . Then for each  $z \in b^*$  we have  $|z|_x = |z|_y = 0$ . Thus  $L_{x=y}$  is infinite.

**Case (ii):**  $x = a^i$  and  $y = b^j$  for two distinct symbols  $a, b$  and  $i, j > 0$ . Then for each  $z$  of the form  $(xy)^n$  we have  $|z|_x = |z|_y = n$ . Thus  $L_{x=y}$  is infinite.

**Case (iii):**  $x = a^i$  for some  $i > 0$  but  $y$  contains two different symbols. Let  $b \in \Sigma$  with  $b \neq a$ . Then for each  $z \in b^*$  we have  $|z|_x = |z|_y = 0$ . Thus  $L_{x=y}$  is infinite.

**Case (iv):**  $x$  and  $y$  both contain two different symbols. Let  $a \in \Sigma^*$ . Then for each  $z \in a^*$  we have  $|z|_x = |z|_y = 0$ . Thus  $L_{x=y}$  is infinite.  $\square$

We could consider the generalization of  $L_{x=y}$  to more than two words:

$$L_{x_1=x_2=\dots=x_n} = \{z \in \Sigma^* : |z|_{x_1} = |z|_{x_2} = \dots = |z|_{x_n}\}.$$

The following examples show that deciding the finiteness of  $L_{x_1=x_2=\dots=x_n}$  for  $n \geq 3$  is more subtle than the case  $n = 2$ . Suppose  $\Sigma = \{0, 1\}$ . Then  $L_{0=1=00=11}$  and  $L_{0=1=01=10}$  are finite languages, but  $L_{00=11=000=111}$  is not.

Consider  $L_{0=1=00=11}$ . For any maximal subword consisting of 0's, the number of 0's exceeds the number of 00's, and similar for 1 and 11. So  $L_{0=1=00=11} = \{\epsilon\}$ .

Consider  $L_{0=1=01=10}$ . Since  $|z|_{01} = |z|_{10}$ , as shown in Figure 1, the words in this language must start and end with the same character. There cannot be a 00 or the number of 0's exceeds that of 01 and 10, and similar for 11. So, the language is a subset of  $(01)^*0 \cup (10)^*1 \cup \{\epsilon\}$ . But no word  $z$  in this language, other than  $\epsilon$ , has  $|z|_0 = |z|_1$ . Therefore,  $L_{0=1=01=10} = \{\epsilon\}$ .

Consider  $L_{00=11=000=111}$ . It contains  $(01)^*$ , and hence is infinite.

Lacking a general condition for finiteness, we prove the following sufficient condition.

**Theorem 8.** *If  $|x_1| = \dots = |x_n|$  then  $L_{x_1=x_2=\dots=x_n}$  is infinite.*

*Proof.* Let  $\ell = |x_1|$ . Consider the cyclic order- $\ell$  de Bruijn word  $w$  of length  $k^\ell$  over the cardinality- $k$  alphabet  $\Sigma$ . Such a word is guaranteed to exist for all  $k \geq 2$  and  $\ell \geq 1$ ; see, e.g., [8]. Let  $w'$  be the prefix of  $w$  of length  $\ell - 1$ . Then  $w^i w' \in L_{x_1=x_2=\dots=x_n}$  for all  $i \geq 1$ .  $\square$

## References

- [1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2nd edition, 2001.
- [2] A. Ehrenfeucht and D. M. Silberger. Periodicity and unbordered segments of words. *Discrete Math.* **26** (1979), 101–109.



- [3] G. Gamard, G. Richomme, J. Shallit, and T. J. Smith. Periodicity in rectangular arrays. *Info. Proc. Letters* **118** (2017), 58–63.
- [4] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.
- [5] M. Lothaire. *Combinatorics on Words*, Vol. 17 of *Encyclopedia of Mathematics and Its Applications*. Addison-Wesley, 1983.
- [6] R. C. Lyndon and M. P. Schützenberger. The equation  $a^M = b^N c^P$  in a free group. *Michigan Math. J.* **9** (1962), 289–298.
- [7] R. J. Parikh. On context-free languages. *J. ACM* **13** (1966), 570–581.
- [8] A. Ralston. De Bruijn sequences — a model example of the interaction of discrete mathematics and computer science. *Math. Mag.* **55** (1982), 131–143.
- [9] J. Shallit. *A Second Course in Formal Languages and Automata Theory*. Cambridge Univ. Press, 2009.